

## TP 2 *MiddleWare* - 2019

Il s'agit d'enrichir l'application permettant de gérer une collection distribuée de morceaux de musique au format mp3 vue au TP1. L'application sera composée d'un ensemble de serveurs, chacun gérant une partie de la collection localisée sur un site (une machine). Le client référencera l'ensemble de ces serveurs, il permettra des recherches sur différents critères. Il s'agira de faire évoluer l'application dans trois directions :

- *streaming* : intégrer le *streaming* dans le client et serveur. Vous êtes libres d'utiliser les technologies de *streaming* que vous voudrez (soyez pragmatiques et restez strictement dans le cadre légal, bien entendu...) mais la librairie *libvlc* offre des fonctionnalités de haut niveau, faciles à mettre en oeuvre, pour le *streaming*. D'autres solutions libres sont possibles, par exemple avec *openCV*.

- l'architecture proposée oblige les clients à gérer le pool de serveurs. Proposer une solution moins lourde pour les clients, dans laquelle un meta-serveur fédère les serveurs disponibles. Tous les clients se connectant au méta serveurs devront avoir accès à l'ensemble des serveurs connectés au méta serveur.

- intégrer le service de messagerie : la découverte de lecteurs et la mise à jour des listes est difficile en mode synchrone. Utiliser *IceStrom* pour notifier l'ensemble des événements qu'un serveur peut supporter et communiquer à ses clients.

### Questions subsidiaires :

1. Y-a-t-il un intérêt à passer par un méta-serveur ? Que faire, dans ce cas, pour limiter le trafic sur le réseau ?
2. Est-ce qu'inclure la vidéo change quelque chose à l'application développée ?
3. Quelle stratégie d'enregistrement des serveurs ? Comment faire pour découvrir à chaud de nouveaux serveurs ? Pour sécuriser l'accès aux serveurs ?